Rate-dependent plasticity: creep and swelling

The classical deviatoric metal creep behavior in Abaqus/Standard:

- can be defined using user subroutine CREEP or by providing parameters as input for some simple creep laws;
- can model either isotropic creep (using Mises stress potential) or anisotropic creep (using Hill's anisotropic stress potential);
- is active only during steps using the coupled temperature-displacement procedure, the transient soils consolidation procedure, and the quasi-static procedure;
- requires that the material's elasticity be defined as linear elastic behavior;
- can be modified to implement the auxiliary creep hardening rules specified in Nuclear Standard NEF 9-5T, "Guidelines and Procedures for Design of Class 1 Elevated Temperature Nuclear System Components"; these rules are exercised by means of a constitutive model developed by Oak Ridge National Laboratory (ORNL – Oak Ridge National Laboratory constitutive model);
- can be used in combination with creep strain rate control in analyses in which the creep strain rate must be kept within a certain range; and
- can potentially result in errors in calculated creep strains if anisotropic creep and plasticity occur simultaneously (discussed below).

Rate-dependent gasket behavior in Abaqus/Standard:

- uses unidirectional creep as part of the model of the gasket's thickness-direction behavior;
- can be defined using user subroutine CREEP or by providing parameters as input for some simple creep laws;
- is active only during steps using the quasi-static procedure; and
- requires that an elastic-plastic model be used to define the rate-independent part of the thickness-direction behavior of the gasket.

Volumetric swelling behavior in Abaqus/Standard:

- can be defined using user subroutine CREEP or by providing tabular input;
- can be either isotropic or anisotropic;
- is active only during steps using the coupled temperature-displacement procedure, the transient soils consolidation procedure, and the quasi-static procedure; and
- requires that the material's elasticity be defined as linear elastic behavior.

The following topics are discussed:

See Also

About the material library

Inelastic behavior

In Other Guides

Defining the gasket behavior directly using a gasket behavior model

*CREEP

*CREEP STRAIN RATE CONTROL

*POTENTIAL

*SWELLING

*RATIOS

Defining a creep law

Defining swelling

Products: Abaqus/Standard Abaqus/CAE

Creep behavior

Creep behavior is specified by the equivalent uniaxial behavior—the creep "law." In practical cases creep laws are typically of very complex form to fit experimental data; therefore, the laws are defined with user subroutine CREEP, as discussed below. Alternatively, five common creep laws are provided in Abaqus/Standard: the power law, the hyperbolic-sine law, the double power law, the Anand law, and the Darveaux law. These standard creep laws are used for modeling secondary or steady-state creep. Creep is defined by including creep behavior in the material model definition (Material data definition). Alternatively, creep can be defined in conjunction with gasket behavior to define the rate-dependent behavior of a gasket.

Input File Usage:

Use the following options to include creep behavior in the material model definition:

```
*MATERIAL
*CREEP
```

Use the following options to define creep in conjunction with gasket behavior:

```
*GASKET BEHAVIOR
*CREEP
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**

Choosing a creep model

The power-law creep model is attractive for its simplicity. However, it is limited in its range of application. The time-hardening version of the power-law creep model is typically recommended only in cases when the stress state remains essentially constant. The strain-hardening version of power-law creep should be used when the stress state varies during an analysis. In the case where the stress is constant and there are no temperature and/or field dependencies, the time-hardening and strain-hardening versions of the power-creep law are equivalent. For either version of the power law, the stresses should be relatively low.

In regions of high stress, such as around a crack tip, the creep strain rates frequently show an exponential dependence of stress. The hyperbolic-sine creep law shows exponential dependence on the stress, σ, at high stress levels ($\sigma/\sigma^0 \gg 1$, where σ^0 is the yield stress) and reduces to the power-law at low stress levels (with no explicit time dependence).

The double power, Anand, and Darveaux models are particularly well suited for modeling the behavior of solder alloys used in electronic packaging and have been shown to produce accurate results for a wide range of temperatures and strain rates.

None of the above models is suitable for modeling creep under cyclic loading. The ORNL model (ORNL – Oak Ridge National Laboratory constitutive model) is an empirical model for stainless steel that gives approximate results for cyclic loading without having to perform the cyclic loading numerically. Generally, creep models for cyclic loading are complicated and must be added to a model with user subroutine CREEP or with user subroutine UMAT.

Modeling simultaneous creep and plasticity

If creep and plasticity occur simultaneously and implicit creep integration is in effect, both behaviors may interact and a coupled system of constitutive equations needs to be solved. If creep and plasticity are isotropic, Abaqus/Standard properly takes into account such coupled behavior, even if the elasticity is anisotropic. However, if creep and plasticity are anisotropic, Abaqus/Standard integrates the creep equations without taking plasticity into account, which may lead to substantial errors in the creep strains. This situation develops only if plasticity and creep are active at the same time, such as would occur during a long-term load increase; one would not expect to have a problem if there is a short-term preloading phase in which plasticity dominates, followed by a creeping phase in which no further yielding occurs. Integration of the creep laws and rate-dependent plasticity are discussed in Rate-dependent metal plasticity (creep).

Time power law and power law models

The time power law and power law models described below are equivalent to the “time hardening” and the “strain hardening” forms but avoid their drawbacks. The time power law and power law models rewrite the laws in such a way that the typical parameter values do not cause numerical difficulties. In addition, the units of all of the parameters are physical, which makes unit conversion easier if it is required.

Time power law model

The time power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left(\frac{\tilde{q}}{q_0} \right)^n \left(\dot{\varepsilon}_0 t \right)^m,$$

where $\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$ are defined above; and $\dot{\varepsilon}_0$, n, m, and q_0 are material parameters.

The model is equivalent to the “time hardening” form. It is recommended that you use the time power law model when the value of the parameter A is very small ($A \le 10^{-27}$). In this case the equivalent time power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage:

```
*CREEP, LAW=TIME POWER LAW
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Time Power**

Power law model

The power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left[\left(\frac{\tilde{q}}{q_0} \right)^n \left[(m+1)\, \bar{\varepsilon}^{cr} \right]^m \right) \right]^{\frac{1}{m+1}},$$

where $\dot{\bar{\varepsilon}}^{cr}$, $\bar{\varepsilon}^{cr}$, and $\tilde{q}$ are defined above; and $\dot{\varepsilon}_0$, n, m, and q_0 are material parameters.

This model is equivalent to the "strain hardening" form. It is recommended that you use the power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage:

```
*CREEP, LAW=POWER LAW
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Power**

Time/Strain hardening models

Time hardening and strain hardening models to specify creep are available. However, to avoid the drawbacks of these models, it is recommended that you use the time power law and power law models (see Time power law and power law models).

Time hardening form

The "time hardening" form is the simpler of the two forms and is defined as

$$\dot{\bar{\varepsilon}}^{cr} = A\tilde{q}^n t^m,$$

where

$\dot{\bar{\varepsilon}}^{cr}$

is the uniaxial equivalent creep strain rate, $\sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}$,

$\tilde{q}$

is the uniaxial equivalent deviatoric stress,

t

is the total or the creep time, and

A*, *n*, and *m

are defined by you as functions of temperature.

$\tilde{q}$ is Mises equivalent stress or Hill's anisotropic equivalent deviatoric stress according to whether isotropic or anisotropic creep behavior is defined (discussed below). For physically reasonable behavior *A* and *n* must be positive and $-1 < m \leq 0$.

Input File Usage:

```
*CREEP, LAW=TIME
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Time-Hardening**

Strain hardening form

The "strain hardening" form is

$$\dot{\bar{\varepsilon}}^{cr} = \left(A\tilde{q}^n\left[(m+1)\,\bar{\varepsilon}^{cr}\right]^m\right)^{\frac{1}{m+1}},$$

where $\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$ are defined above and $\bar{\varepsilon}^{cr}$ is the equivalent creep strain.

Input File Usage:

```
*CREEP, LAW=STRAIN
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Strain-Hardening**

Numerical difficulties

Depending on the choice of units for either form, the value of *A* may be very small for typical creep strain rates. If *A* is less than 10^{-27}, numerical difficulties can cause errors in the material calculations. Therefore, use another system of units or use the time power law or power law model (described below) to avoid such difficulties in the calculation of creep strain increments.

Time-dependent behavior

In the "time hardening" form and the time power law model, the total time or the creep time can be used. The total time is the accumulated time over all general analysis steps. The creep time is the sum of the times of the procedures with time-dependent material behavior. If the total time is used, it is recommended that small step times compared to the creep time be used for any steps for which creep is not active in an analysis; this is necessary to avoid changes in hardening behavior in subsequent steps.

Input File Usage:

Use one of the following options:

```
*CREEP, TIME=TOTAL (default)
*CREEP, TIME=CREEP
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Time: Total** or **Creep**

Hyperbolic-sine law model

The hyperbolic-sine law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A(\sinh B\tilde{q})^n \exp\left(-\frac{\triangle H}{R\left(\theta - \theta^Z\right)}\right),$$

where

$\dot{\bar{\varepsilon}}^{cr}$ **and** $\tilde{q}$

are defined above,

θ

is the temperature,

θ^Z

is the user-defined value of absolute zero on the temperature scale used,

$\triangle H$

is the activation energy,

R

is the universal gas constant, and

A*, *B*, and *n

are other material parameters.

This model includes temperature dependence, which is apparent in the above expression; however, the parameters *A*, *B*, *n*, $\triangle H$, and *R* cannot be defined as functions of temperature.

Input File Usage:

Use both of the following options:

```
*CREEP, LAW=HYPERB
*PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z
```

Abaqus/CAE Usage:

Define both of the following:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Hyperbolic-Sine**

Any module: **Model** > **Edit Attributes** > ***model_name***: **Absolute zero temperature**

Numerical difficulties

As with the power law, *A* may be very small for typical creep strain rates. If *A* is very small (such as less than 10^{-27}), use another system of units to avoid numerical difficulties in the calculation of creep strain increments.

Anand model

The Anand model is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A\left[\sinh\left(\xi\frac{\tilde{q}}{s}\right)\right]^{\frac{1}{m}} \exp\left(-\frac{Q}{R\left(\theta - \theta^Z\right)}\right),$$

where

$\dot{\bar{\varepsilon}}^{cr}$**,** $\tilde{q}$**, *R*,** θ**, and** θ^Z

are defined above,

Q

is the activation energy,

s

is the deformation resistance, and

A, m, and ξ

are material parameters.

The evolution equation for the deformation resistance, s (initially $s = s_0$), is

$$\dot{s} = h_0 \left|1 - \frac{s}{s^*}\right|^a sign\left(1 - \frac{s}{s^*}\right)\dot{\bar{\varepsilon}}^{cr},$$

with

$$s^* = \hat{s}\left[\frac{1}{A}\dot{\bar{\varepsilon}}^{cr}\exp\left(\frac{Q}{R\left(\theta - \theta^Z\right)}\right)\right]^n,$$

where

$$h_0 = A_0 + A_1\left(\theta - \theta^Z\right) + A_2\left(\theta - \theta^Z\right)^2 + A_3\dot{\bar{\varepsilon}}^{cr} + A_4\left(\dot{\bar{\varepsilon}}^{cr}\right)^2,$$

and a, n, $\hat{s}$, A_0, A_1, A_2, A_3, and A_4 are material parameters.

In addition, the initial deformation resistance is a function of temperature of the form

$$s_0 = S_1 + S_2\left(\theta - \theta^Z\right) + S_3\left(\theta - \theta^Z\right)^2,$$

where S_1, S_2, and S_3 are material parameters.

Input File Usage:

Use both of the following options:

```
*CREEP, LAW=ANAND
*PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z
```

Abaqus/CAE Usage:

Define both of the following:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Anand**

Any module: **Model** > **Edit Attributes** > ***model_name***: **Absolute zero temperature**

Darveaux model

The Darveau model involves both primary and secondary creep. The secondary creep (steady-state) component is described by a standard hyperbolic sine law

$$\dot{\bar{\varepsilon}}^{cr}_s = C_{ss}[\sinh(\alpha\tilde{q})]^n \exp\left(-\frac{Q}{R\left(\theta - \theta^Z\right)}\right).$$

The steady-state law is modified to include the primary creep effects through

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\bar{\varepsilon}}^{cr}_s\left[1 + \epsilon_T B\exp\left(-B\dot{\bar{\varepsilon}}^{cr}_s t\right)\right],$$

where

$\dot{\bar{\varepsilon}}^{cr}$, $\tilde{q}$, **R, Q,** θ, and θ^Z

are defined above,

C_{ss}

is the steady-state creep prefactor,

α

is the steady-state creep power law breakdown, and

n, ϵ_T, and B

are other material parameters.

Input File Usage:

Use both of the following options:

*CREEP, LAW=DARVEAUX
*PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage:

Define both of the following:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Darveaux**

Any module: **Model** > **Edit Attributes** > ***model_name***: **Absolute zero temperature**

Double power model

The double power law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A_1 \exp\left(-\frac{B_1}{(\theta - \theta^Z)}\right)\left(\frac{\tilde{q}}{\sigma_0}\right)^{C_1} + A_2 \exp\left(-\frac{B_2}{(\theta - \theta^Z)}\right)\left(\frac{\tilde{q}}{\sigma_0}\right)^{C_2},$$

where

$\dot{\bar{\varepsilon}}^{cr}$, $\tilde{q}$, θ, **and** θ^Z

are defined above,

σ_0

is the normalized stress, and

A_1, A_2, B_1, B_2, C_1, **and** C_2

are other material parameters.

Input File Usage:

Use both of the following options:

*CREEP, LAW=DOUBLE POWER
*PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage:

Define both of the following:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Law: Double Power**

Any module: **Model** > **Edit Attributes** > ***model_name***: **Absolute zero temperature**

Anisotropic creep

Anisotropic creep can be defined to specify the stress ratios that appear in Hill's function. You must define the ratios R_{ij} in each direction that will be used to scale the stress value when the creep strain rate is calculated. The ratios can be defined as constant or dependent on temperature and other predefined field variables. The ratios are defined with respect to the user-defined local material directions or the default directions (see Orientations). Further details are provided in Hill anisotropic yield/creep. Anisotropic creep is not available when creep is used to define a rate-dependent gasket behavior since only the gasket thickness-direction behavior can have rate-dependent behavior.

Input File Usage:

*POTENTIAL

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Creep**: **Suboptions** > **Potential**

Volumetric swelling behavior

As with the creep laws, volumetric swelling laws are usually complex and are most conveniently specified in user subroutine CREEP as discussed below. However, a means of tabular input is also provided for the form

$$\dot{\bar{\varepsilon}}^{sw} = f(\theta, f_1, f_2, \ldots),$$

where $\dot{\bar{\varepsilon}}^{sw}$ is the volumetric strain rate caused by swelling and f_1, f_2, $\ldots$ are predefined fields such as irradiation fluxes in cases involving nuclear radiation effects. Up to six predefined fields can be specified.

Volumetric swelling cannot be used to define a rate-dependent gasket behavior.

Input File Usage:

```
*SWELLING
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Swelling**

Anisotropic swelling

Anisotropy can easily be included in the swelling behavior. If anisotropic swelling behavior is defined, the anisotropic swelling strain rate is expressed as

$$\dot{\bar{\varepsilon}}^{sw}_{A} = \dot{\bar{\varepsilon}}^{sw}_{11} + \dot{\bar{\varepsilon}}^{sw}_{22} + \dot{\bar{\varepsilon}}^{sw}_{33} = (r_{11} + r_{22} + r_{33})\,\frac{1}{3}\dot{\bar{\varepsilon}}^{sw},$$

where $\dot{\bar{\varepsilon}}^{sw}$ is the volumetric swelling strain rate that you define either directly (discussed above) or in user subroutine CREEP. The ratios r_{11}, r_{22}, and r_{33} are also user-defined. The directions of the components of the swelling strain rate are defined by the local material directions, which can be either user-defined or the default directions (see Orientations).

Input File Usage:

Use both of the following options:

```
*SWELLING
*RATIOS
```

Abaqus/CAE Usage:

Property module: material editor: **Mechanical** > **Plasticity** > **Swelling**: **Suboptions** > **Ratios**

User subroutine CREEP

User subroutine CREEP provides a very general capability for implementing viscoplastic models such as creep and swelling models in which the strain rate potential can be written as a function of equivalent pressure stress, p; the Mises or Hill's equivalent deviatoric stress, $\tilde{q}$; and any number of solution-dependent state variables. Solution-dependent state variables are used in conjunction with the constitutive definition; their values evolve with the solution and can be defined in this subroutine. Examples are hardening variables associated with the model.

The user subroutine can also be used to define very general rate- and time-dependent thickness-direction gasket behavior. When an even more general form is required for the strain rate potential, user subroutine UMAT (User-defined mechanical material behavior) can be used.

Input File Usage:

Use one or both of the following options. Only the first option can be used to define gasket behavior.

```
*CREEP, LAW=USER
*SWELLING, LAW=USER
```

Abaqus/CAE Usage:

Use one or both of the following models. Only the first model can be used to define gasket behavior.

Property module: material editor:
Mechanical > **Plasticity** > **Creep**: **Law: User defined**
Mechanical > **Plasticity** > **Swelling**: **Law: User subroutine CREEP**

Removing creep effects in an analysis step

You can specify that no creep (or viscoelastic) response can occur during certain analysis steps, even if creep (or viscoelastic) material properties have been defined.

Input File Usage:

Use one of the following options:

```
*COUPLED TEMPERATURE-DISPLACEMENT, CREEP=NONE
*SOILS, CONSOLIDATION, CREEP=NONE
```

Abaqus/CAE Usage:

Use one of the following options:

Step module: **Create Step**:
Coupled temp-displacement: toggle off **Include creep/swelling/ viscoelastic behavior**

Soils: **Pore fluid response: Transient consolidation**: toggle off **Include creep/swelling/viscoelastic behavior**

Integration

Explicit integration, implicit integration, or both integration schemes can be used in a creep analysis, depending on the procedure used, the parameters specified for the procedure, the presence of plasticity, and whether or not geometric nonlinearity is requested.

Application of explicit and implicit schemes

Nonlinear creep problems are often solved efficiently by forward-difference integration of the inelastic strains (the "initial strain" method). This explicit method is computationally efficient because, unlike implicit methods, iteration is not required. Although this method is only conditionally stable, the numerical stability limit of the explicit operator is usually sufficiently large to allow the solution to be developed in a small number of time increments.

Abaqus/Standard uses either an explicit or an implicit integration scheme or switches from explicit to implicit in the same step. These schemes are outlined first, followed by a description of which procedures use these integration schemes.

1. Integration scheme 1: Starts with explicit integration and switches to implicit integration based on either stability or if plasticity is active. The stability limit used in explicit integration is discussed in the next section.
2. Integration scheme 2: Starts with explicit integration and switches to implicit integration when plasticity is active. The stability criterion does not play a role here.
3. Integration scheme 3: Always uses implicit integration.

The use of the above integration schemes is determined by the procedure type, your choice of the integration type to be used, as well as whether or not geometric nonlinearity is requested. For quasi-static and coupled temperature-displacement procedures, if you do not choose an integration type, integration scheme 1 is used for a geometrically linear analysis and integration scheme 3 is used for a geometrically nonlinear analysis. You can force Abaqus/Standard to use explicit integration for creep and swelling effects in coupled temperature-displacement or quasi-static procedures, when plasticity is not active throughout the step (integration scheme 2). Explicit integration can be used regardless of whether or not geometric nonlinearity has been requested (see General and perturbation procedures).

For a transient soils consolidation procedure, the implicit integration scheme (integration scheme 3) is always used, irrespective of whether a geometrically linear or nonlinear analysis is performed.

Input File Usage:

Use one of the following options to restrict Abaqus/Standard to using explicit integration:

```
*VISCO, CREEP=EXPLICIT
*COUPLED TEMPERATURE-DISPLACEMENT, CREEP=EXPLICIT
```

Abaqus/CAE Usage:

Use one of the following options to restrict Abaqus/Standard to using explicit integration:

Step module: **Create Step**:
Visco: **Incrementation**: **Creep/swelling/viscoelastic integration: Explicit**

Coupled temp-displacement: toggle on **Include creep/swelling/ viscoelastic behavior**: **Incrementation**: **Creep/swelling/viscoelastic integration: Explicit**

Automatic monitoring of stability limit during explicit integration

Abaqus/Standard monitors the stability limit automatically during explicit integration. If, at any point in the model, the creep strain increment $\left(\dot{\bar{\varepsilon}}^{cr}\big|_t \Delta t\right)$ is larger than the total elastic strain, the problem will become unstable. Therefore, a stable time step, Δt_s, is calculated every increment by

$$\Delta t_s = 0.5 \frac{\varepsilon^{el}\big|_t}{\dot{\bar{\varepsilon}}^{cr}\big|_t},$$

where $\varepsilon^{el}\big|_t$ is the equivalent total elastic strain at time *t*, the beginning of the increment, and $\dot{\bar{\varepsilon}}^{cr}\big|_t$ is the equivalent creep strain rate at time *t*. Furthermore,

$$\varepsilon^{el}\big|_t = \frac{\tilde{q}\,\big|_t}{\widetilde{E}},$$

where $\tilde{q}\,\big|_t$ is the Mises stress at time *t*, and

$$\widetilde{E} = 2\left(1+\nu\right)\left(\mathbf{n} : \mathbf{D}^{el} : \mathbf{n}\right) \approx 2.5\overline{E},$$

where

$\mathbf{n} = \partial \tilde{q}|_t / \partial \boldsymbol{\sigma}$

is the gradient of the deviatoric stress potential,

$\mathbf{D}^{el}$

is the elasticity matrix, and

$\overline{E}$

is an effective elastic modulus—for isotropic elasticity $\overline{E}$ can be approximated by Young's modulus.

At every increment for which explicit integration is performed, the stable time increment, Δt_s, is compared to the critical time increment, Δt_c, which is calculated as follows:

$$\Delta t_c = \frac{errtol}{\dot{\bar{\varepsilon}}^{cr}\big|_{t+\Delta t} - \dot{\bar{\varepsilon}}^{cr}\big|_t}.$$

The quantity *errtol* is an error tolerance that you define as discussed below. If Δt_s is less than Δt_c, Δt_s is used as the time increment, which would mean that the stability criterion was limiting the size of the time step further than required by accuracy considerations. Abaqus/Standard will automatically switch to the backward difference operator (the implicit method, which is unconditionally stable) if Δt_s is less than Δt_c for nine consecutive increments, you have not restricted Abaqus/Standard to explicit integration as discussed above, and there is sufficient time left in the analysis (time left $\geq 50\Delta t$). The stiffness matrix will be reformed at every iteration if the implicit algorithm is used.

Specifying the tolerance for automatic incrementation

The integration tolerance must be chosen so that increments in stress, $\triangle\boldsymbol{\sigma}$, are calculated accurately. Consider a one-dimensional example. The stress increment, $\triangle\sigma$, is

$$\triangle\sigma = E\triangle\varepsilon^{el} = E\left(\triangle\varepsilon - \triangle\varepsilon^{cr}\right),$$

where $\triangle\varepsilon^{el}$, $\triangle\varepsilon$, and $\triangle\varepsilon^{cr}$ are the uniaxial elastic, total, and creep strain increments, respectively, and *E* is the elastic modulus. For $\triangle\sigma$ to be calculated accurately, the error in the creep strain increment, $\triangle\varepsilon^{cr}_{err}$, must be small compared to $\triangle\varepsilon^{el}$; that is,

$$\triangle\varepsilon^{cr}_{err} \ll \triangle\varepsilon^{el}.$$

Measuring the error in $\triangle\varepsilon^{cr}$ as

$$\triangle\varepsilon^{cr}_{err} = \left(\dot{\bar{\varepsilon}}^{cr}\big|_{t+\triangle t} - \dot{\bar{\varepsilon}}^{cr}\big|_t\right)\triangle t$$

leads to

$$\left(\dot{\bar{\varepsilon}}^{cr}\big|_{t+\triangle t} - \dot{\bar{\varepsilon}}^{cr}\big|_t\right)\triangle t \ll \triangle\varepsilon^{el} = \frac{\triangle\sigma}{E}\text{ , or}$$

$$errtol \ll \frac{\triangle\sigma}{E}.$$

You define *errtol* for the applicable procedure by choosing an acceptable stress error tolerance and dividing this by a typical elastic modulus; therefore, it should be a small fraction of the ratio of the typical stress and the effective elastic modulus in a problem. It is important to recognize that this approach for selecting a value for *errtol* is often very conservative, and acceptable solutions can usually be obtained with higher values.

Input File Usage:

Use one of the following options:

```
*VISCO, CETOL=errtol
*COUPLED TEMPERATURE-DISPLACEMENT, CETOL=errtol
*SOILS, CONSOLIDATION, CETOL=errtol
```

Abaqus/CAE Usage:

Use one of the following options:

Step module: **Create Step**:
Visco: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Coupled temp-displacement: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Soils: **Pore fluid response: Transient consolidation**: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Loading control using creep strain rate

In superplastic forming a controllable pressure is applied to deform a body. Superplastic materials can deform to very large strains, provided that the strain rates of the deformation are maintained within very tight tolerances. The objective of the superplastic analysis is to predict how the pressure must be controlled to form the component as fast as possible without exceeding a superplastic strain rate anywhere in the material.

To achieve this using Abaqus/Standard, the controlling algorithm is as follows. During an increment Abaqus/Standard calculates $r_{\max}$, the maximum value of the ratio of the equivalent creep strain rate to the target creep strain rate for any integration point in a specified element set. If $r_{\max}$ is less than 0.2 or greater than 3.0 in a given increment, the increment is abandoned and restarted with the following load modifications:

$$\begin{aligned} r_{\max} < 0.2 \;\; p = 2.0\, p_{\text{old}},\ \text{or} \\ r_{\max} > 3.0 \;\; p = 0.5\, p_{\text{old}}, \end{aligned}$$

where *p* is the new load magnitude and p_{old} is the old load magnitude. If $0.2 \le r_{\max} \le 3.0$, the increment is accepted; and at the beginning of the following time increment, the load magnitudes are modified as follows:

$$\begin{aligned} 0.2 \le r_{\max} < 0.5 &\quad p = 1.5\, p_{\text{old}}; \\ 0.5 \le r_{\max} < 0.8 &\quad p = 1.2\, p_{\text{old}}; \\ 0.8 \le r_{\max} < 1.5 &\quad p = p_{\text{old}}\,;\ \text{or} \\ 1.5 \le r_{\max} \le 3.0 &\quad p = p_{\text{old}}/1.2. \end{aligned}$$

When you activate the above algorithm, the loading in a creep and/or swelling problem can be controlled on the basis of the maximum equivalent creep strain rate found in a defined element set. As a minimum requirement, this method is used to define a target equivalent creep strain rate; however, if required, it can also be used to define the target creep strain rate as a function of equivalent creep strain (measured as log strain), temperature, and other predefined field variables. The creep strain dependency curve at each temperature must always start at zero equivalent creep strain.

A solution-dependent amplitude is used to define the minimum and maximum limits of the loading (see Defining a solution-dependent amplitude for superplastic forming analysis). Any number or combination of loads can be used. The current value of $r_{\max}$ is available for output as discussed below.

Input File Usage:

Use all of the following options:

```
*AMPLITUDE, NAME=name, DEFINITION=SOLUTION DEPENDENT
*CLOAD, *DLOAD, *DSLOAD, and/or *BOUNDARY with
AMPLITUDE=name
*CREEP STRAIN RATE CONTROL, AMPLITUDE=name, ELSET=elset
```

The *AMPLITUDE option must appear in the model definition portion of an input file, while the loading options (*CLOAD, *DLOAD, *DSLOAD, and *BOUNDARY) and the *CREEP STRAIN RATE CONTROL option should appear in each relevant step definition.

Abaqus/CAE Usage:

Creep strain rate control is not supported in Abaqus/CAE.

Elements

Rate-dependent plasticity (creep and swelling behavior) can be used with any continuum, shell, membrane, gasket, and beam element in Abaqus/Standard that has displacement degrees of freedom. Creep (but not swelling) can also be defined in the thickness direction of any gasket element in conjunction with the gasket behavior definition.

Output

In addition to the standard output identifiers available in Abaqus/Standard (Abaqus/Standard output variable identifiers), the following variables relate directly to creep and swelling models:

CEEQ

Equivalent creep strain, $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}\, dt$.

CESW

Magnitude of swelling strain.

The following output, which is relevant only for an analysis with creep strain rate loading control as discussed above, is printed at the beginning of an increment and is written automatically to the results file and output database file when any output to these files is requested:

RATIO

Maximum value of the ratio of the equivalent creep strain rate to the target creep strain rate, r_{max}.

AMPCU

Current value of the solution-dependent amplitude.